What is claimed is:

1. In a data packet transmission system, a method of filtering an incoming data packet comprising the steps of:

creating a decision graph having a tree-like hierarchy for a set of filters such that the filters are subdivided and grouped along edges and stages of the tree-like hierarchy according to common filter values;

receiving a data packet at a point of ingress to the data packet transmission system;

comparing filters along edges of the tree-like hierarchy to portions of the data packet to determine where there are matches; and

if matches are made along an entire edge of the tree-like hierarchy, filtering the data packet stream according to number of condition values compared.

2. The method according to claim 1, wherein said creating step is performed so that the tree-like hierarchy includes wildcards at specified byte values.

3. The method according to claim 2, wherein said creating step is performed by the step of subdividing the set of filters according to common values at specified bytes of said filters.

4. The method according to claim 3, wherein the comparing step further comprises the step of dynamically choosing which byte of a filter to compare against portions of the classification key.

5. The method according to claim 4, further comprising the step of choosing a next byte to compare based on the number of wildcards in a filter.

6. The method according to claim 5, further comprising the step of choosing the byte with the most wildcards.

7. The method according to claim 1, further comprising the step of classifying the data packet according to the number of condition value matches encountered during comparison with the tree-like hierarchy.

8. The method according to claim 1, wherein the condition values are limited by a number of bytes in a filter within the groups of filters.

9. The method according to claim 1, wherein a size of the tree-like hierarchy is minimized based on a specified criteria.

10. The method according to claim 9, wherein the specified criteria is based on a number of irrelevant condition values in the data packet stream.

11. The method according to claim 1, wherein the step of creating a decision tree is performed such that the set of filters are divided into groups of filters based on specific bytes in a search key.

12. The method according to claim 11, wherein the set of filters are divided based on comparing one byte at a time in the search key.

13. The method according to claim 1, wherein the filter conditions are divided into groups of filters based on a maximum number of bits which all filters in the groups of filters have in common.

14. The method according to claim 1, further comprising:

storing intermediate matching values; and

matching the stored results with a next byte of the data packet.

15. In connection with a router that receives incoming packets from a data network, a method of filtering the incoming packets at the interface of the router to provide differentiated services, the method comprising the steps of:

constructing a decision tree structure that stores a set of filters in a tree-like hierarchy, the tree-like hierarchy including edges and stages, the set of filters subdivided according to their values along the edges and stages of the tree-like hierarchy;

the interface of the router receiving at least one incoming data packet; and

comparing segments of the incoming data packet to values of the filters stored in the tree-like hierarchy such that the segments are compared along stages and not every edge of the tree-like hierarchy to allow the incoming data packet to be differentiated according to matches between the segments compared and specific filter values.

16. The method of claim 15 wherein said comparison step further comprises the step of executing a single instruction within the PPU that filters the incoming data packet according to its content.

17. The method of claim 16 wherein the step of executing the single instruction further comprises the step of advancing from the beginning of the incoming data packet by a predetermined amount of offset before beginning to compare.

18. The method of claim 15 wherein said comparing step comprises the further steps of:

comparing a first specific byte of the incoming data packet to values of filters in the upper most stage of the tree-like hierarchy;

once a match is made, comparing a second specific byte of the incoming data packet to values of filters contained in a stage below the upper most stage of the tree-like hierarchy and along edges that connect to filters in the upper most stage that match in values to the first specific byte.

19. The method of claim 18 further comprising the step of comparing further bytes of the incoming data packets to values of filters contained in subsequent stages of the tree-like structure until a match is made at the lowest stage of the tree-like structure or no match at all.

20. The method of claim 19 wherein said comparing steps further comprise the step of dynamically choosing the next byte of the incoming data packet to compare.

21. The method of claim 19 wherein said comparing steps further comprise the step of replacing specified values in the filters with wildcards.

22. The method of claim 19 wherein said comparing steps further comprise the step of collapsing values in any stage of the tree-like structure so that multiple branches from a parent can lead to the same child.

23. A router for use in a data network and for allowing the implementation of differentiated services by filtering incoming data packets, the router comprising:

an interface to the data network;

a packet processing unit coupled to said interface and adapted for receiving incoming data packets via the interface from the data network;

memory means operably coupled to said packet processing unit and adapted for storing software instructions that cause said packet processing unit to filter incoming data packets; and

a data structure for storing a set of filters in a tree-like hierarchy, the tree-like hierarchy including edges organized into stages, the set of filters subdivided according to their values along the edges and stages of the tree-like hierarchy;

wherein said software instructions are further adapted to cause said packet processing unit to compare segments of incoming data packets to values of the filters stored in the tree-like hierarchy such that the segments are compared along stages and not every edge of the tree-like hierarchy to allow the incoming data packet to be differentiated according to matches between the segments compared and specific filter values.

24. The router of claim 23 wherein said packet processing unit is an application specific integrated circuit or general purpose CPU.

25. The router of claim 24 wherein said application specific integrated circuit is further adapted to filter incoming IP packets based on the contents of the IP packets and using said tree-like structure to provide differentiated services.

26. The router of claim 23 wherein said software instructions comprise a single instruction capable of causing said packet processing unit to filter incoming data packets.

27. The router of claim 23 wherein said software instructions are further adapted to cause said packet processing unit to compare specific bytes of incoming data packets with filter values of said tree-like structure.

28. The router of claim 27 wherein said software instructions are further adapted to cause said packet processing unit to compare a first specific byte of the incoming data packet to values of filters in the upper most stage of the tree-like hierarchy and once a match is made to compare a second specific byte of the incoming data packet to values of filters contained in a stage below the upper most stage of the tree-like hierarchy and along edges that connect to filters in the upper most stage that match in values to the first specific byte.